**Lines of CODE Metrics**

Lines of code metrics is used to determine the size of a program based on the number of lines in the source code. This metric is useful to predict the complexity of a program and the amount of effort a programmer will need to tackle developing and maintaining the code.

I was unable to use the Lines of Code metric in the metrics reloaded plug in due to an error that me and many of my colleagues encountered. However, after searching online I saw a recommendation to the “Statistic” plug in that would allow me to attain the same result.

My lab teacher recommended my group to pick a package and run the plug in there instead of the whole project. I chose this directory: src/main/java/org/jabref/preferences.

After doing some research, I encountered that in this book “Refactoring in Large Software Projects” by Martin Lippert and Stephen Roock there is rule called “Rule of 30” that states: A class should contain an average of less than 30 methods, resulting in up to 900 lines of code. The data that I collect in the package falls under this rule except for the JabRefPreferences class which has 2130 lines of code. This is a big amount of code for a single class raising a major red flag that could be challenging for a person to understand and refactor later down the line.

Also, there is a long parameter list in the GuiPreferences, the refactoring of this code smell will lead to a creation of a new class lowering the number of lines of code in this class but in turn will probably lead to the increase in total lines of code due to the newly created class.